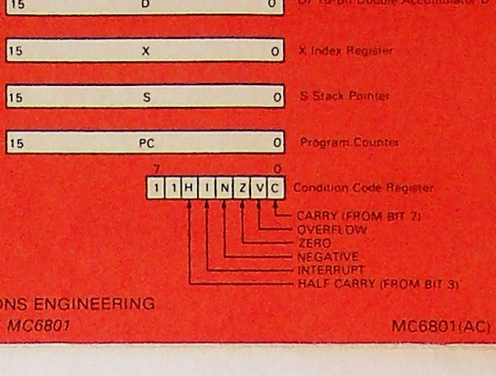
MC6801 / 68701 / 6803 MICROCOMPUTER

INSTRUCTION SET SUMMARY



MOTOROLA INC.

Integrated Circuits Division

3501 Ed Bluestein Blvd.
Austin, Texas 78721
Telephone: (512) 928-6000

NMOS MICROCOMPUTER APPLICATIONS ENGINEERING

**MC68701 IS THE EPROM VERSION OF MC6801*

MC6801(AC)

MC6801 REGISTER MAP

7	6	5	4	3	2	1	0	
								$00 Port 1 Direction
								$01 Port 2 Direction
								$02 Port 1 I/O Data
								$03 Port 2 I/O Data
								$04 Port 3 Direction
								$05 Port 4 Direction
								$06 Port 3 I/O Data
								$07 Port 4 I/O Data
ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL	$08 Timer Control/Status
								$09 Timer Counter Hi Byte
								$0A Timer Counter Low Byte
								$0B Timer Output Compare Hi
								$0C Timer Output Compare Low
								$0D Timer Input Capture Hi
								$0E Timer Input Capture Low
IS3 F	IS3 E		OSS	LTCH				$0F I/O Port 3 Control/Status
				CC1	CC0	S1	S0	$10 Serial Rate/Mode
RDRF	ORFE	TDRE	RIE	RE	TIE	TE	WU	$11 Serial Control/Status
								$12 Serial Rx Data
								$13 Serial Tx Data
SB	RAME					PGE	PLE	$14 RAM/EROM Control
								$15 To $1F = Reserved

ASCII CHARACTER SET (7-BIT CODE)

MS CHAR	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
LS CHAR								
0 0000	NUL	DLE	SP	0	@	P	`	p
1 0001	SOH	OC1	!	1	A	Q	a	q
2 0010	STX	DC2	"	2	B	R	b	r
3 0011	ETX	DC3	#	3	C	S	c	s
4 0100	EOT	DC4	$	4	D	T	d	t
5 0101	ENQ	NAK	%	5	E	U	e	u
6 0110	ACK	SYN	&	6	F	V	f	v
7 0111	BEL	ETB	'	7	G	W	g	w
8 1000	BS	CAN	(	8	H	X	h	x
9 1001	HT	EM	)	9	I	Y	i	y
A 1010	LF	SUB	*	:	J	Z	j	z
B 1011	VT	ESC	+	;	K	[	k	{
C 1100	FF	FS	,	<	L	\	l	|
D 1101	CR	GS	-	=	M	]	m	}
E 1110	SO	RS	.	>	N	↑	n	~
F 1111	SI	US	/	?	O	←	o	DEL

INTERRUPT MEMORY ASSIGNMENT

	VECTOR (FIRST) MS BYTE	LOCATION (SECOND) LS BYTE	INTERRUPT
HIGHEST PRIORITY	FFFE	FFFF	RESTART
	FFFC	FFFD	NON-MASKABLE INTERRUPT
	FFFA	FFFB	SWI
	FFF8	FFF9	IRQ1/INPUT STROBE 3 e.g. IS3
	FFF6	FFF7	IRQ2/TIMER INPUT CAPTURE
	FFF4	FFF5	IRQ2/TIMER OUTPUT COMPARE
	FFF2	FFF3	IRQ2/TIMER OVERFLOW
LOWEST PRIORITY	FFF0	FFF1	IRQ2/SERIAL COMMUNICATIONS INTERUPT

FIRST INSTR ADDR FORMED BY FETCHING 2 BYTES FROM PER MEM. ASSIGN

MC6801 MODE SELECTS

MODE	PROGRAM CONTROL (PC2)	(PC1)	(PC0)	ROM	RAM	INTERRUPT VECTORS	BUS	
	PINS 10	9	8					
7	Hi	Hi	Hi	I	I	I	I	Single Chip Expanded Multiplexed
6	Hi	Hi	Lo	I	I	I	(3)Ep M	Expanded Non-Multiplexed
5	Hi	Lo	Hi	I	I	I	(3) Ep	Single Chip Test
4	Hi	Lo	Lo	I(2)	R(1)	I	I	64K Address/ D
3	Lo	Hi	Hi	E	E	E	(4)Ep M	Ports 3 & 4
2	Lo	Hi	Lo	E	E	E	(4)Ep M	External
1	Lo	Lo	Hi	I	I	E	(4)Ep M	Test Data Out put From ROM & RAM To I/O Port 3
0	Lo	Lo	Lo	I	I	E	(4)Ep M	

Ep — EXPANDED
E — EXTERNAL all vectors are external
I — INTERNAL
M — MULTIPLEXED
* — First two addresses read from external after reset
(1) Address for RAM XX80 XXFF
(2) ROM Disabled
(3) Address lines may be substituted for I/O (inputs only)
(4) Full 64K address only

INSTRUCTION MAP

OP	MNEM	MODE	#	#
00	*	INHER	2	1
01	NOP		1	1
02	*		1	1
03	*		1	1
04	LSRD	INHER	3	1
05	ASLD		3	1
06	TAP		2	1
07	TPA		2	1
08	INX		3	1
09	DEX		3	1
0A	CLV		2	1
0B	SEV		2	1
0C	CLC		2	1
0D	SEC		2	1
0E	CLI		2	1
0F	SEI		2	1
10	SBA		2	1
11	CBA		2	1
12	*		1	1
13	*		1	1
14	*		1	1
15	*		1	1
16	TAB	INHER	2	1
17	TBA		2	1
18	*		1	1
19	DAA	INHER	2	1
1A	*		1	1
1B	ABA	INHER	2	1
1C	*		1	1
1D	*		1	1
1E	*		1	1
1F	*		1	1
20	BRA	REL	3	2
21	BRN		3	2
22	BHI		3	2
23	BLS		3	2
24	BCC		3	2
25	BCS		3	2
26	BNE		3	2
27	BEQ		3	2
28	BVC		3	2
29	BVS		3	2
2A	BPL		3	2
2B	BMI		3	2
2C	BGE		3	2
2D	BLT		3	2
2E	BGT		3	2
2F	BLE	REL	3	2
30	TSX	INHER	3	1
31	INS		3	1
32	PULA		4	1
33	PULB		4	1

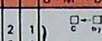
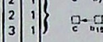
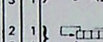
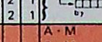
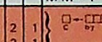
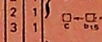
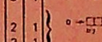
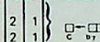
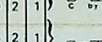
OP	MNEM	MODE	#	#
34	DES	INHER	3	1
35	TXS		3	1
36	PSHA		3	1
37	PSHB		3	1
38	PULX		5	1
39	RTS		5	1
3A	ABX		3	1
3B	RTI		10	1
3C	PSHX		4	1
3D	MUL		10	1
3E	WAI		9	1
3F	SWI		12	1
40	NEGA		2	1
41	*			
42	*			
43	COMA	INHER	2	1
44	LSRA		2	1
45	*			
46	RORA	INHER	2	1
47	ASRA		2	1
48	ASLA		2	1
49	ROLA		2	1
4A	DECA		2	1
4B	*			
4C	INCA		2	1
4D	TSTA		2	1
4E	*			
4F	CLRA	INHER	2	1
50	NEGB		2	1
51	*			
52	*			
53	COMB	INHER	2	1
54	LSRB		2	1
55	*			
56	RORB		2	1
57	ASRB		2	1
58	ASLB		2	1
59	ROLB		2	1
5A	DECB		2	1
5B	*			
5C	INCB	INHER	2	1
5D	TSTB		2	1
5E	*			
5F	CLRB	INHER	2	1
60	NEG	INDXD	6	2
61	*			
62	*			
63	COM	INDXD	6	2
64	LSR	INDXD	6	2
65	*			
66	ROR	INDXD	6	2
67	ASR		6	2

OP	MNEM	MODE	#	#
68	ASL	INDXD	6	2
69	ROL		6	2
6A	DEC		6	2
6B	*			
6C	INC	INDXD	6	2
6D	TST		6	2
6E	JMP		3	2
6F	CLR		6	2
70	NEG	EXTND	6	3
71	*			
72	*			
73	COM	EXTND	6	3
74	LSR		6	3
75	*			
76	ROR	EXTND	6	3
77	ASR		6	3
78	ASL		6	3
79	ROL		6	3
7A	DEC		6	3
7B	*			
7C	INC	EXTND	6	3
7D	TST		6	3
7E	JMP		3	3
7F	CLR		6	3
80	SUBA	IMMED	2	2
81	CMPA		2	2
82	SBCA		2	2
83	SUBD		4	3
84	ANDA		2	2
85	BITA		2	2
86	LDAA		2	2
87	*			
88	EORA	IMMED	2	2
89	ADCA		2	2
8A	ORAA		2	2
8B	ADDA		2	2
8C	CPX		4	3
8D	BSR	REL	6	2
8E	LDS	IMMED	3	3
8F	*			
90	SUBA	DIR	3	2
91	CMPA		3	2
92	SBCA		3	2
93	SUBD		5	2
94	ANDA		3	2
95	BITA		3	2
96	LDAA		3	2
97	STAA		3	2
98	EORA		3	2
99	ADCA		3	2
9A	ORAA		3	2
9B	ADDA		3	2

OP	MNEM	MODE	#	#
9C	CPX	DIR	5	2
9D	JSR		5	2
9E	LDS	DIR	4	2
9F	STS		4	2
A0	SUBA	INDXD	4	2
A1	CMPA		4	2
A2	SBCA		4	2
A3	SUBD		6	2
A4	ANDA		4	2
A5	BITA		4	2
A6	LDAA		4	2
A7	STAA		4	2
A8	EORA		4	2
A9	ADCA		4	2
AA	ORAA		4	2
AB	ADDA		4	2
AC	CPX		6	2
AD	JSR		6	2
AE	LDS		5	2
AF	STS		5	2
B0	SUBA	EXTND	4	3
B1	CMPA		4	3
B2	SBCA		4	3
B3	SUBD		6	3
B4	ANDA		4	3
B5	BITA		4	3
B6	LDAA		4	3
B7	STAA		4	3
B8	EORA		4	3
B9	ADCA		4	3
BA	ORAA		4	3
BB	ADDA		4	3
BC	CPX		6	3
BD	JSR		6	3
BE	LDS		5	3
BF	STS		5	3
C0	SUBB	IMMED	2	2
C1	CMPB		2	2
C2	SBCB		2	2
C3	ADDD		4	3
C4	ANDB	IMMED	2	2
C5	BITB		2	2
C6	LDAB		2	2
C7	*			
C8	EORB		2	2
C9	ADCB		2	2
CA	ORAB		2	2
CB	ADDB		2	2
CC	LDD		3	3
CD	*			
CE	LDX	IMMED	3	3
CF	*			

OP	MNEM	MODE	#	#
D0	SUBB	DIR	3	2
D1	CMPB		3	2
D2	SBCB		3	2
D3	ADDD		5	2
D4	ANDB		3	2
D5	BITB		3	2
D6	LDAB		3	2
D7	STAB		3	2
D8	EORB		3	2
D9	ADCB		3	2
DA	ORAB		3	2
DB	ADDB		3	2
DC	LDD		4	2
DD	STD		4	2
DE	LDX	DIR	4	2
DF	STX		4	2
E0	SUBB	INDXD	4	2
E1	CMPB		4	2
E2	SBCB		4	2
E3	ADDD		6	2
E4	ANDB		4	2
E5	BITB		4	2
E6	LDAB		4	2
E7	STAB		4	2
E8	EORB		4	2
E9	ADCB		4	2
EA	ORAB		4	2
EB	ADDB		4	2
EC	LDD		5	2
ED	STD		5	2
EE	LDX		5	2
EF	STX		5	2
F0	SUBB	EXTND	4	3
F1	CMPB		4	3
F2	SBCB		4	3
F3	ADDD		6	3
F4	ANDB		4	3
F5	BITB		4	3
F6	LDAB		4	3
F7	STAB		4	3
F8	EORB		4	3
F9	ADCB		4	3
FA	ORAB		4	3
FB	ADDB		4	3
FC	LDD		5	3
FD	STD		5	3
FE	LDX		5	3
FF	STX		5	3

*UNDEFINED OP CODE

ACCUMULATOR & OPERATIONS	MEM NMN	IMMED OP	~	#	DIRECT OP	~	#	INDEX OP	~	#	EXTEND OP	~	#	INHER OP	~	#	BOOLEAN EXPRESSION	CONDITION CODES H	I	N	Z	V	C
Add Acmltrs	ABA													1B	2	1	A + B → A	↕	•	↕	↕	↕	↕
Add B to X	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Add with Carry	ADCA	89	2	2	99	3	2	A9	4	2	B9	4	3				A + M + C → A	↕	•	↕	↕	↕	↕
	ADCB	C9	2	2	D9	3	2	E9	4	2	F9	4	3				B + M + C → B	↕	•	↕	↕	↕	↕
Add	ADDA	8B	2	2	9B	3	2	AB	4	2	BB	4	3				A + M → A	↕	•	↕	↕	↕	↕
	ADDB	CB	2	2	DB	3	2	EB	4	2	FB	4	3				B + M → A	↕	•	↕	↕	↕	↕
Add Double	ADDD	C3	4	3	D3	5	2	E3	6	2	F3	6	3				D + M:M + 1 → D	•	•	↕	↕	↕	↕
And	ANDA	84	2	2	94	3	2	A4	4	2	B4	4	3				A · M → A	•	•	↕	↕	R	•
	ANDB	C4	2	2	D4	3	2	E4	4	2	F4	4	3				B · M → B	•	•	↕	↕	R	•
Shift Left, Arithmetic	ASL							68	6	2	78	6	3					•	•				
	ASLA													48	2	1		•	•				
	ASLB													58	2	1		•	•				
Shift Left Dbl	ASLD													05	3	1		•	•				
Shift Right, Arithmetic	ASR							67	6	2	77	6	3					•	•				
	ASRA													47	2	1		•	•				
	ASRB													57	2	1		•	•				
Bit Test	BITA	85	2	2	95	3	2	A5	4	2	B5	4	3				A · M	•	•	↕	↕	R	•
	BITB	C5	2	2	D5	3	2	E5	4	2	F5	4	3				B · M	•	•	↕	↕	R	•
Compare Acmltrs	CBA													11	2	1	A - B	•	•	↕	↕	↕	↕
Clear	CLR							6F	6	2	7F	6	3				00 → M	•	•	R	S	R	R
	CLRA													4F	2	1	00 → A	•	•	R	S	R	R
	CLRB													5F	2	1	00 → B	•	•	R	S	R	R
Compare	CMPA	81	2	2	91	3	2	A1	4	2	B1	4	3				A - M	•	•	↕	↕	↕	↕
	CMPB	C1	2	2	D1	3	2	E1	4	2	F1	4	3				B - M	•	•	↕	↕	↕	↕
1's Complement	COM							63	6	2	73	6	3				$\overline{M}$ → M	•	•	↕	↕	R	S
	COMA													43	2	1	$\overline{A}$ → A	•	•	↕	↕	R	S
	COMB													53	2	1	$\overline{B}$ → B	•	•	↕	↕	R	S
Decimal Adj. A	DAA													19	2	1	Adj binary sum to BCD	•	•	↕	↕	↕	↕
Decrement	DEC							6A	6	2	7A	6	3				M - 1 → M	•	•	↕	↕	↕	•
	DECA													4A	2	1	A - 1 → A	•	•	↕	↕	↕	•
	DECB													5A	2	1	B - 1 → B	•	•	↕	↕	↕	•
Exclusive OR	EORA	88	2	2	98	3	2	A8	4	2	B8	4	3				A⊕M → A	•	•	↕	↕	R	•
	EORB	C8	2	2	D8	3	2	E8	4	2	F8	4	3				B⊕M → B	•	•	↕	↕	R	•
Increment	INC							6C	6	2	7C	6	3				M + 1 → M	•	•	↕	↕	↕	•
	INCA													4C	2	1	A + 1 → A	•	•	↕	↕	↕	•
	INCB													5C	2	1	B + 1 → B	•	•	↕	↕	↕	•
Load Acmltrs	LDAA	86	2	2	96	3	2	A6	4	2	B6	4	3				M → A	•	•	↕	↕	R	•
	LDAB	C6	2	2	D6	3	2	E6	4	2	F6	4	3				M → B	•	•	↕	↕	R	•
Load Double	LDD	CC	3	3	DC	4	2	EC	5	2	FC	5	3				M:M + 1 → D	•	•	↕	↕	R	•
Logical Shift, Left	LSL							68	6	2	78	6	3					•	•				
	LSLA													48	2	1		•	•				
	LSLB													58	2	1		•	•				
	LSLD													05	3	1		•	•				
Shift Right, Logical	LSR							64	6	2	74	6	3					•	•	R			
	LSRA													44	2	1		•	•	R			
	LSRB													54	2	1		•	•	R			
	LSRD													04	3	1		•	•	R			
Multiply	MUL													3D	10	1	A * B → D	•	•	•	•	•	↕
2's Complement (Negate)	NEG							60	6	2	70	7	3				00 - M → M	•	•	↕	↕	↕	↕
	NEGA													40	2	1	00 - A → A	•	•	↕	↕	↕	↕
	NEGB													50	2	1	00 - B → B	•	•	↕	↕	↕	↕
No Operation	NOP													01	2	1	PC + 1 → PC	•	•	•	•	•	•
Inclusive OR	ORAA	8A	2	2	9A	3	2	AA	4	2	BA	4	3				A + M → A	•	•	↕	↕	R	•
	ORAB	CA	2	2	DA	3	2	EA	4	2	FA	4	3				B + M → B	•	•	↕	↕	R	•
Push Data	PSHA													36	3	1	A → Stack	•	•	•	•	•	•
	PSHB													37	3	1	B → Stack	•	•	•	•	•	•
Pull Data	PULA													32	4	1	Stack → A	•	•	•	•	•	•
	PULB													33	4	1	Stack → B	•	•	•	•	•	•
Rotate Left	ROL							69	6	2	79	6	3					•	•				
	ROLA													49	2	1		•	•				
	ROLB													59	2	1		•	•				
Rotate Right	ROR							66	6	2	76	6	3					•	•				
	RORA													46	2	1		•	•				
	RORB													56	2	1		•	•				
Subtract Acmltr	SBA													10	2	1	A - B → A	•	•	↕	↕	↕	↕
Subtract with Carry	SBCA	82	2	2	92	3	2	A2	4	2	B2	4	3				A - M - C → A	•	•	↕	↕	↕	↕
	SBCB	C2	2	2	D2	3	2	E2	4	2	F2	4	3				B - M - C → B	•	•	↕	↕	↕	↕
Store Acmltrs	STAA				97	3	2	A7	4	2	B7	4	3				A → M	•	•	↕	↕	R	•
	STAB				D7	3	2	E7	4	2	F7	4	3				B → M	•	•	↕	↕	R	•
	STD				DD	4	2	ED	5	2	FD	5	3				D → M:M + 1	•	•	↕	↕	R	•
Subtract	SUBA	80	2	2	90	3	2	A0	4	2	B0	4	3				A - M → A	•	•	↕	↕	↕	↕
	SUBB	C0	2	2	D0	3	2	E0	4	2	F0	4	3				B - M → B	•	•	↕	↕	↕	↕
Subtract Double	SUBD	83	4	3	93	5	2	A3	6	2	B3	6	3				D - M:M + 1 → D	•	•	↕	↕	↕	↕
Transfer Acmltr	TAB													16	2	1	A → B	•	•	↕	↕	R	•
	TBA													17	2	1	B → A	•	•	↕	↕	R	•
Test, Zero or Minus	TST							6D	6	2	7D	6	3				M - 00	•	•	↕	↕	R	R
	TSTA													4D	2	1	A - 00	•	•	↕	↕	R	R
	TSTB													5D	2	1	B - 00	•	•	↕	↕	R	R

JUMP & BRANCH OPERATIONS	NMN	RELATIVE OP	~	#	DIRECT OP	~	#	INDEX OP	~	#	EXTEND OP	~	#	INHER. OP	~	#	BRANCH TEST	CONDITION CODES H	I	N	Z	V	C
Branch Always	BRA	20	3	2													None	•	•	•	•	•	•
Branch If Carry Clear	BCC	24	3	2													C = 0	•	•	•	•	•	•
Branch If Carry Set	BCS	25	3	2													C = 1	•	•	•	•	•	•
Branch If = Zero	BEQ	27	3	2													Z = 1	•	•	•	•	•	•
Branch If ≥ Zero	BGE	2C	3	2													N⊕V = 0	•	•	•	•	•	•
Branch If > Zero	BGT	2E	3	2													Z + (N⊕V) = 0	•	•	•	•	•	•
Branch If Higher	BHI	22	3	2													C + Z = 0	•	•	•	•	•	•
Branch If ≤ Zero	BLE	2F	3	2													Z + (N⊕V) = 1	•	•	•	•	•	•
Branch If Lower or Same	BLS	23	3	2													C + Z = 1	•	•	•	•	•	•
Branch If < Zero	BLT	2D	3	2													N⊕V = 1	•	•	•	•	•	•
Branch If Minus	BMI	2B	3	2													N = 1	•	•	•	•	•	•
Branch If Not Equal Zero	BNE	26	3	2													Z = 0	•	•	•	•	•	•
Branch If Overflow Clear	BVC	28	3	2													V = 0	•	•	•	•	•	•
Branch If Overflow Set	BVS	29	3	2													V = 1	•	•	•	•	•	•
Branch If Plus	BPL	2A	3	2													N = 0	•	•	•	•	•	•
Branch Never	BRN	21	3	2													None	•	•	•	•	•	•
Branch If Higher or Same	BHS	24	3	2													C = 0	•	•	•	•	•	•
Branch If Lower	BLO	25	3	2													C = 1	•	•	•	•	•	•
Branch to Subroutine	BSR	8D	6	2														•	•	•	•	•	•
Jump	JMP							6E	3	2	7E	3	3					•	•	•	•	•	•
Jump to Subroutine	JSR				9D	5	2	AD	6	2	BD	6	3					•	•	•	•	•	•
No Operation	NOP													01	2	1		•	•	•	•	•	•
Return from Interrupt	RTI													3B	10	1		↕	↕	↕	↕	↕	↕
Return from Subroutine	RTS													39	5	1		•	•	•	•	•	•
Software Interrupt	SWI													3F	12	1		•	S	•	•	•	•
Wait for Interrupt	WAI													3E	9	1		•	•	•	•	•	•

INDEX REGISTER OPERATIONS	NMN	IMMED OP	~	#	DIRECT OP	~	#	INDEX OP	~	#	EXTEND OP	~	#	INHER OP	~	#	BOOLEAN EXPRESSION	CONDITION CODES H	I	N	Z	V	C
Compare Index Register	CPX	8C	4	3	9C	5	2	AC	6	2	BC	6	3				X - M:M + 1	•	•	↕	↕	↕	↕
Decrement Index Register	DEX													09	3	1	X - 1 → X	•	•	•	↕	•	•
Increment Index Register	INX													08	3	1	X + 1 → X	•	•	•	↕	•	•
Load Index Register	LDX	CE	3	3	DE	4	2	EE	5	2	FE	5	3				M:M + 1 → X	•	•	↕	↕	R	•
Store Index Register	STX				DF	4	2	EF	5	2	FF	5	3				X → M:M + 1	•	•	↕	↕	R	•
Add B to Index Register	ABX													3A	3	1	00:B + X → X	•	•	•	•	•	•
Push Index Register	PSHX													3C	4	1	X → Stack	•	•	•	•	•	•
Pull Index Register	PULX													38	5	1	Stack → X	•	•	•	•	•	•
Transfer X to SP	TXS													35	3	1	X → SP	•	•	•	•	•	•
Transfer SP to X	TSX													30	3	1	SP → X	•	•	•	•	•	•

STACK POINTER OPERATIONS	NMN	IMMED OP	~	#	DIRECT OP	~	#	INDEX OP	~	#	EXTEND OP	~	#	INHER OP	~	#	BOOLEAN EXPRESSION	CONDITION CODES H	I	N	Z	V	C
Decrement Stack Pointer	DES													34	3	1	SP - 1 → SP	•	•	•	•	•	•
Increment Stack Pointer	INS													31	3	1	SP + 1 → SP	•	•	•	•	•	•
Load Stack Pointer	LDS	8E	3	3	9E	4	2	AE	5	2	BE	5	3				M:M + 1 = SP	•	•	↕	↕	R	•
Store Stack Pointer	STS				9F	4	2	AF	5	2	BF	5	3				SP → M:M + 1	•	•	↕	↕	R	•
Transfer X to SP	TXS													35	3	1	X → SP	•	•	•	•	•	•
Transfer SP to X	TSX													30	3	1	SP → X	•	•	•	•	•	•

CONDITION CODE REGISTER OPERATIONS	NMN	INHER OP	~	#	BOOLEAN OPERATION	CONDITION CODES H	I	N	Z	V	C
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	R	•
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	S	•
Acmltr A → CCR	TAP	06	2	1	A → CCR	↕	↕	↕	↕	↕	↕
CCR → Acmltr A	TPA	07	2	1	CCR → A	•	•	•	•	•	•

LEGEND:

OP	Operation Code (Hexadecimal)	H	Half-Carry from Bit 3
~	Number of MPU Cycles	I	Interrupt Mask
#	Number of Program Bytes	N	Negative (Sign Bit)
+	Arithmetic Plus	Z	Zero (Byte)
−	Arithmetic Minus	V	Overflow, 2's Complement
·	Boolean AND	C	Carry from Bit 7
MSP	Contents of Memory Location Pointed to be Stack Pointer	R	Reset Always
		S	Set Always
+	Boolean Inclusive OR	↕	Test and Set if True, Cleared Otherwise
⊕	Boolean Exclusive OR	•	Not Affected
$\overline{M}$	Complement of M	CCR	Condition Code Register
→	Transfer Into	LS	Least Significant
0	Bit = Zero	MS	Most Significant
00	Byte = Zero		

12000 PRINTED IN USA 5-79 IMPERIAL LITHO 87934 25M MC6800